Topic: Intel® Xeon™ Processor (DP and MP) Signal Integrity Models Usage Guidelines, v1. 6

From: Intel Corporation

To: OEM and Intel signal integrity model users

Date: March, 2002

1.0 Contents of this Document

1.0 Contents of this Document 1
2.0 Release Information 1
2.1 Contents 1
3.0 Intel® Xeon™ Processor Signal Integrity Model Usage 3
3.1 Introduction 3
3.2 Design Space Exploration 3
3.3 Socket, Interposer, and Package Modeling 4
3.4 Test Load Correction 5
3.5 Source Synchronous Timing Simulation 6
3.6 Platform Timing Assumptions 7
3.7 Model Characterization 10
4.0 Validation 11
5.0 Package Model Information 12
5.1 Model Description: 12
5.2 Model Limitations/Assumptions : 13
5.3 Model Usage: 13
5.4 Package Model File Details 14
5.5 Model Units: 15
6.0 Disclaimer and Legal Information 16

2.0 Release Information

2.1 Contents

All contents are preliminary and subject to change.

File	Description
Signal_integrity_models_usage_guidelines_r1_5.pdf	This document
cpu_x.txt	Package parasitic files in a non-vendor specific format. The “x” in the file name ranges from 1 to 6
cpu_endx.ibs	I/O end agent buffer model in IBIS 3.1 format. “x” ranges from 1 to 5.
cpu_midx.ibs	I/O mid agent buffer model in IBIS 3.1 format. “x” ranges from 1 to 5.
Length Adjustment Worksheet r1_5.xls	Microsoft Excel* spreadsheet to aid in length matching, including package length compensation and package signal lengths.
gowsim_nt.exe	Overshoot checker for Windows* NT* workstations
gowsim_sun	Overshoot checker for Sun* Microsystems Solaris
gowsim_ibm	Overshoot checker for IBM* AIX*
gowsim_hp	Overshoot checker for Hewlett-Packard* HP-UX*
Readme.pdf	Overshoot checker application note

Revision History	
Revision	**Description**
Revision 0.5	Revision 0.5 models are based on simulations of the target I/O buffer design and are accurate with respect to the frozen buffer design, pre-layout.
Revision 0.6	Revision 0.6 incorporates the final package models and a length adjustment worksheet. The IBIS models are almost identical to the rev0.5 models, with only a few error corrections. A spot check on the rev0.5 models has determined that these models are still worst case. Actual silicon may slightly improve edge rates with respect to the model 1(reduce edge rate) and model 7 (increase edge rate), while maintaining drive strength equal to rev0.5 models.
Revision 1.0	Revision 1.0 updates the IBIS models for the end and middle agents. These new models are based on extracted design data and have had some limited correlation to early silicon measurements. Also included are updated correlation waveforms based on the revision 1.0 models. The number of models has been reduced due to tuning of the I/O buffers and refinement of the buffer characterization process.
Revision 1.1	Revision 1.1 updates the cpu_mid3.ibs and cpu_mid4.ibs files to correct an error in the [voltage range] field.
Revision 1.2	Revision 1.2 updates the cpu_endx.ibs models to adhere to the IBIS spec for number of points in the curves and ensuring VT curves begin at time 0ns. This Usage Guidelines document has also been updated to better explain how to build the package model, update the timing correction based on the new waveforms, updated the Length Adjustment Worksheet with package line lengths for the Intel® Xeon™ Processor 31mm package, and remove reference to files that are not included. The shifted models used for source synchronous timing simulations have not been updated since rev0.5. These models are still accurate and still represent the worst case skew possible between strobe and an associated signal.
Revision 1.3	Models updated for Vcc of 1.7V.
Revision 1.4	Models updated for buffers of production Intel® Xeon™ Processor steppings.
Revision 1.41	No changes to the models. Only changes to the length matching spreadsheet.
Revision 1.42	Contains both 1.8V models and scaled down 1.7V models. The P-keeper in the AGTL+ buffer for the middle agent will now shut off after 2.5 ns during the low to high transition instead of after 5.0 ns. This modification is required to reflect changes in C-step parts which eliminate excessive overshoot on data signals.
Revision 1.5	Section 2.1 File contents updated to include overshoot checker utilities.
Revision 1.6	Updated contents for Intel® Xeon™ Processor MP.

3.0 Intel® Xeon™ Processor (DP and MP) Signal Integrity Model Usage

3.1 Introduction

This section describes the models and simulation methods required to verify that Intel® Xeon™ Processor (DP and MP) system bus meets signal integrity and common clock timing requirements. It is assumed that the user is skilled in simulation setup, results analysis, and translation of models from the IBIS format[†]. The intent is to enable the user to create simulation runs that stress timing and noise margins as prescribed in the *Intel® Xeon™ Processor at 1.40 GHz, 1.50 GHz. 1.70 GHz and 2 GHz Datasheet*, and the *Intel ® Xeon ™ Processor MP at 1.40 GHz, 1.50 GHz and 1.60 GHz*.

Due to narrow timing and noise margins associated with the processor system bus, traditional Fast/Slow Signal Integrity Models are not sufficient to model all worst case conditions. To effectively stimulate phenomena such as ringback or undershoot, certain combinations of pull-up and pull-down strengths, edge rate, I/O pad capacitance, and package characteristics must be present. The driver models needed to create worst case behavior are listed below. The characteristics of each model are listed in the appendix for reference. Models 1-4 were created with Vcc at 1.80V, and a temperature of 0°C. Model 5 was created with Vcc at 1.50V, and a temperature of 100°C.

All end agent models contain on-die termination defined in the driver shunt statement. An external termination resistor should not be declared.

Package models are supplied individually, and should be matched with the appropriate driver in simulation to create the necessary corner conditions. The package models represent the nominal trace geometry, high and low impedance geometry, and the maximum cross talk condition. Please see Section 5.0 Package Model for a detailed description of how to include the package models in simulation.

3.2 Design Space Exploration

To understand the sensitivity and stability of a system under design, Monte Carlo simulation methods should be employed. In the Monte Carlo simulation, all systematic parameters are varied randomly within their tolerance ranges. Variables such as dielectric constant, trace impedance, socket parasitics, and line length tolerances should be included in the Monte Carlo simulation runs. For initial system characterization, runs of approximately 1000 permutations may be sufficient. Data analysis should focus on determining the input parameter to output result correlation for system variables. The sensitivity of each system variable can then be assessed, and values necessary for creating corner conditions can be determined. By running Monte Carlo simulations during the initial design phase, the designer will gain an increased understanding of the relationships between system variables and system performance.

The following table describes the driver/package models recommended for these Monte Carlo simulation runs. The end and middle agent drivers are named cpu_endx.ibs and cpu_midx.ibs respectively, where ‘x’ is replaced with the correct model number shown in the tables. For example, middle agent driver model 4 refers to “cpu_mid4.ibs”.

Monte Carlo				
Simulation corner	End agent driver	Middle agent driver	Package model	Package length
Full randomized parameter sweep	1 and 5	1 and 5	Nominal	Nominal

[†] The information contained in the IBIS format models is correct. Intel has noted cases of commercial conversion software failing to properly import IBIS data to native simulator formats. Please contact your tool vendor if you experience this problem.

Once system corners have been located, it is recommended that system parameter sweep simulations be run with each of the driver/package model combinations listed in the following tables. For each of the worst case simulation corners, a combination of end and middle agent drivers, and package parameters is provided. In some cases, worst case behavior will be found only by distributing high/low impedance package models among the agents in the system.

Although simulations are targeted at specific worst case corners, it is recommended that all timing and signal integrity metrics be checked on every simulation. To achieve optimal coverage, systematic parameters should be swept for each timing/signal integrity corner listed.

Common Clock Flight Time Corners				
Simulation corner	End agent driver	Middle agent driver	Package model	Package length
Minimum rise time	3	3	Low Z	Short
Maximum rise time	5	5	High Z	Long
Minimum fall time	1	1	Low Z	Short
Maximum fall time	5	5	High Z	Long

Signal Integrity Corners				
Simulation corner	End agent driver	Middle agent driver	Package model	Package length
Overshoot	3	3	Low Z	Short
Undershoot	1	1	Low Z	Short
High side ringback	4	4	High Z	Long
Low side ringback	4	4	High Z	Long

Crosstalk				
Simulation corner	End agent driver	Middle agent driver	Package model	Package length
Max crosstalk	1 or 3	1 or 3	Max crosstalk	Long

3.3 Socket, Interposer, and Package Modeling

Use the package signal line lengths provided in the Intel® Xeon™ Processor (DP and MP) *Length Adjustment Worksheet* in conjunction with the RLC matrix provided in the package models to properly model the processor package. The processor package also contains trace routing on an interposer. This interposer should be modeled and added to the signal simulation in addition to the package models provided. The interposer is routed as symmetric stripline with an impedance range of 45Ω - 55Ω. The trace length on the interposer, including the via, is 100mils. There is virtually no crosstalk contribution from the interposer routing.

The Intel® Xeon™ Processor (DP and MP) 603-pin socket (PGA-603) is modeled with discrete inductors, capacitors and resistors as shown in the following figure. This is a per-signal per-socket model and should be replicated for each signal on each processor.

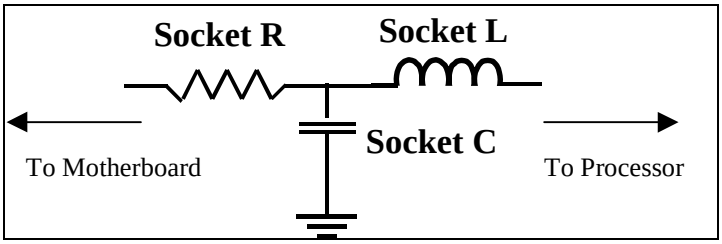


Figure 1. Intel® Xeon™ Processor Socket Model

The following table provides parameter value ranges to be used in the socket model. Vary the value of each socket parameter to create the worst case for the corner under simulation.

Intel® Xeon™ Processor (DP and MP) **(603-Pin) Socket Models**			
Socket Type	Socket L	Socket R	Socket C
Processor	1.0 – 4.3nH	0.005 – 0.020 ohms	1.0 – 1.5pF

.

3.4 Test Load Correction

System flight times must be corrected to eliminate double counting of the driver rise/fall time, and to correct device timings for system loading effects. **It is necessary to subtract the rising/falling delay of the buffer driving into the standard test load from the driver-to-receiver propagation delay in the system.** The processor driver models have been driven into the standard tester load, and the following flight time correction factors have been measured. For reference, the following figure describes the standard test load. For each simulation run, the correction factor corresponding to the active driver should be subtracted from the measured flight time. Proper use of the correction factor requires that simulation time zero begins at the moment of voltage change at the driver pad. If this is not the case, the user should define a correction factor based on the standard load provided.

Flight time correction factors for IBIS models (from Innoveda* XTK* simulation)					
		End Agent		Middle Agent	
Model	Vref	Rising correction (ns)	Falling correction (ns)	Rising correction (ns)	Falling correction (ns)
Model 1	1.20	0.54	0.70	0.57	0.47
Model 2	1.20	0.57	0.72	0.58	0.51
Model 3	1.20	0.40	0.48	0.41	0.33
Model 4	1.20	0.43	0.50	0.43	0.36
Model 5	1.00	0.43	0.58	0.45	0.42
Model f	1.10	0.62	0.41	0.52	0.35
Model fs	1.07	0.82	0.58	0.69	0.50
Model s	0.93	0.55	0.38	0.47	0.36
Model ss	0.96	0.43	0.28	0.37	0.25

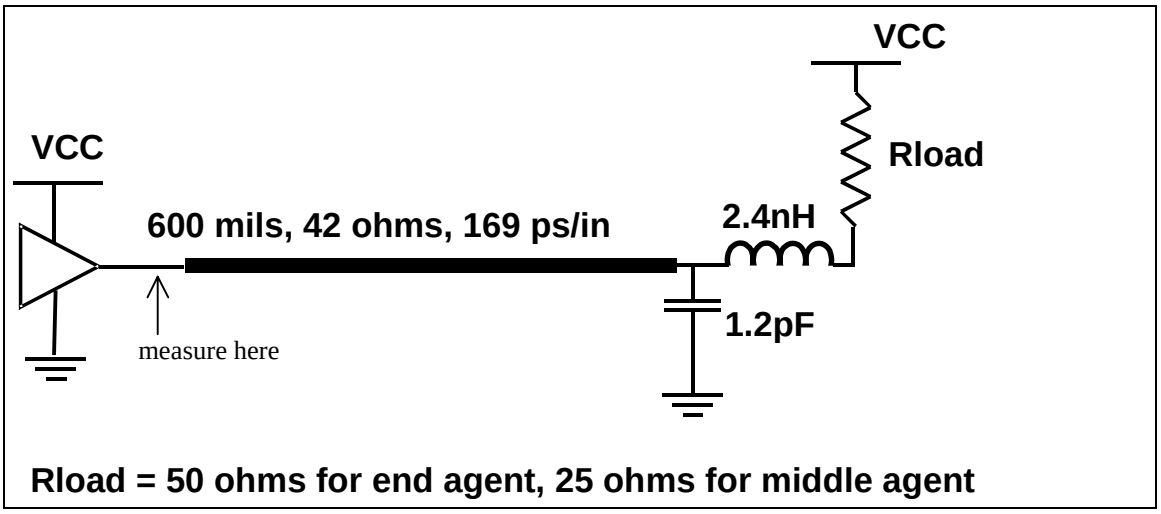


Figure 2. Standard Test Load for Time to Vm Correction.

3.5 Source Synchronous Timing Simulation

Source synchronous bus operation depends on electrical similarity between data and strobe signal paths. Deviations of board impedance, driver/receiver pad capacitance, etc. cause setup and hold margin at the receiver to be eroded from the Tvb/Tva present at the driver. Flight time skew simulations should be run to find the worst case timing difference between data and strobe signals in a single system.

To capture the worst case variation between two signals within a strobe group, the concept of a "shifted" simulation corner is important to understand. In system simulation, a "corner" represents a set of parameters that create the worst case behavior on the bus. Because source synchronous timing budgets are only concerned with the relative flight times of signals within a single system, it is necessary to determine the amount of performance shift that may be seen between signals. This shift is determined by the degree to which parameters within a single system may vary, rather than the total parameter variation across a high volume manufacturing process. For flight time skew analysis, data signals will be assumed to lie in the simulation corner, and strobe signals will be shifted toward nominal conditions, by amounts which are reasonable to expect within a single system.

The electrical differences between data and strobe signals within a single system must be characterized in order to set up flight time skew simulations correctly. Monte Carlo simulation will help the designer to find the conditions that create the slowest and fastest system behavior (see section "Design Space Exploration"). Once these cases are known, shifted simulations can be set up which shift behavior toward nominal from the extreme cases identified. For example, assume it has been determined that high board impedance causes slow behavior, and that the variation of trace impedance within a single manufactured system can be as much as 3 ohms. Across all systems, impedance may range from 45 to 55 ohms. To simulate the maximum possible flight time difference, the slow simulation would use the upper impedance limit (55 ohms), while the shifted simulation would use the value that causes faster behavior (52 ohms).

This approach should be used for all sensitive system variables[1]. The designer must determine the conditions that create fast and slow behavior within the system, and also understand the single system variations of the sensitive parameters. Once this information is in hand, fast/fast shifted and slow/slow shifted simulations can be set up to determine worst-case flight time skew.

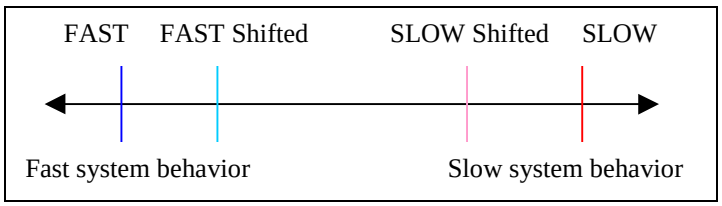


Figure 3. Relative Flight Time Speed

To simulate source synchronous flight time skew, the correct combinations of simulations must be run. There are two basic timing conditions that must be analyzed:

Setup Skew: Data arrives late relative to the associated strobe signal. Data simulation is slow, while strobe simulation is slow shifted. Skew, in this case, is measured as the maximum data flight time minus the minimum strobe flight time[2].

Hold Skew: Data arrives early relative to the associated strobe signal. Data simulation is fast, while strobe simulation is fast shifted. Skew, in this case, is measured as the minimum data flight time minus the maximum strobe flight time.

[1] Package length variation, die capacitance, and driver edge rates are typically the dominant variables affecting flight time skew. Circuit board dielectric constant and etch width variations between data and strobe signals will likely be small, but have the potential to negatively impact timing margins.

[2] In all cases, strobe flight times are measured on the signal's falling edge.

The shifted driver models (fs and ss) capture the maximum same-silicon variation expected within a source synchronous strobe group[3], relative to models f and s. The table below describes driver and package models to be used in flight time skew simulations. All sensitive system parameters should be included and varied within their range to determine worst case skew (variances should only be those that can occur within a single system). **Remember to subtract the appropriate flight time correction factors from the measured simulation flight times when performing flight time skew calculations.**

Intel® Xeon™ Processor (DP and MP) Source Synchronous Flight Time Skew				
Simulation corner	End agent driver	Middle agent driver	Package model	Package length
Slow	S	S	High Z	Maximum Data or Address signal package length
Slow Shifted	Ss	Ss	High Z	Associated Strobe length
Fast	F	F	Low Z	Minimum Data or Address signal package length
Fast Shifted	Fs	Fs	Low Z	Associated Strobe length

3.6 Platform Timing Assumptions

The follow section details the timing assumptions Intel has used to determine its recommended solution space.

Source Synchronous Timings

In a source synchronous bus the clock (or strobe) is driven from the same source as the signal it will sample. The strobe and the signal both propagate to the receiver via the PCB. The receiver then uses the strobe to sample the signal. This eliminates the need to account for worst case flight times and, in theory, will dramatically increase the maximum bus speed.

Setup Time

Equation 1 gives the total loop equation for a source synchronous setup timing requirement.

$$T_{co}(strobe) + T_{flight}(strobe) - T_{setup} - T_{margin\_setup} - T_{co}(data) - T_{flight}(data) = 0$$

Equation 1. Source synchronous setup time

- T_{co}(strobe)[(data)] is the driver delay of the strobe [data] at the transmitting device pad
- T_{flight}(strobe)[(data)] is the flight time of the strobe [data] interconnect
- T_{setup} is the receiver's setup requirement at the receiving device pad
- T_{margin} is the available timing margin for the setup time

The loop equation can be simplified and solved for $T_{margin\_setup.}$ The equation can be broken into 2 parts, valid before and interconnect skew, and then the setup margin can be determined. . This is shown in equations 2 and 3. The Source synchronous setup margin is shown in equation 4.

$$T_{vb} = T_{co}(data)_{\max} - T_{co}(strobe)_{\min}$$

Equation 2. Source synchronous, valid before

$$T_{skew,\max} = T_{flight}(data)_{\max} - T_{flight}(strobe)_{\min}$$

Equation 3. Source synchronous, interconnect skew

$$T_{margin\_setup} = -T_{vb} - T_{skew,\max} - T_{setup}$$

Equation 4. Source synchronous setup margin

[3] Model "fs" refers to cpu_endfs.ibs and cpu_midfs.ibs. Model "ss" refers to cpu_endss.ibs and cpu_midss.ibs

The timing parameters needed to complete this equation are as follows. Timings other than those provided in the component specifications are assumptions that are based on expected platform performance.

- T_{vb} & T_{setup} are specified in the respective component specifications (*Intel® Xeon™ Processor at 1.40 GHz, 1.50 GHz. 1.70 GHz and 2 GHz Datasheet* or the *Intel ® Xeon ™ Processor MP at 1.40 GHz, 1.50 GHz and 1.60 GHz Datasheet,* and chipset datasheet)
- $T_{skew,\ max}$ = 0.59 ns.

Hold Time

Equation 5 gives the total loop equation for a source synchronous hold timing requirement.

$$T_{co}(data) + T_{flight}(data) - T_{margin\_hold} - T_{hold} - T_{flight}(strobe) - T_{co}(strobe) = 0$$

Equation 5. Source synchronous loop equation for hold timing diagram

- T_{co}(strobe)[(data)] is the driver delay of the strobe [data] at the transmitting device pad
- T_{flight}(strobe)[(data)] is the flight time of the strobe [data] interconnect
- T_{hold} is the receiver's setup requirement at the receiving device pad
- T_{margin} is the available timing margin for the setup time

The loop equation can be simplified and solved for $T_{margin\_hold.}$ The equation can be broken into 2 parts, valid before and interconnect skew, as shown in Equation 6 and Equation 7. The hold margin is shown in Equation 8.

$$T_{va} = T_{co}(data)_{\min} - T_{co}(strobe)_{\max}$$

Equation 6. Source synchronous, valid after

$$T_{skew,\min} = T_{flight}(data)_{\min} - T_{flight}(strobe)_{\max}$$

Equation 7. Source synchronous, interconnect skew

$$T_{margin\_hold} = T_{va,\min} + T_{skew,\min} - T_{hold}$$

Equation 8. Source synchronous, hold margin

The timing parameters needed to complete this equation are as follows. Timings other than those provided in the component specifications are assumptions that are based on expected platform performance.

- T_{va} & T_{hold} are specified in the respective component specifications (*Intel® Xeon™ Processor at 1.40 GHz, 1.50 GHz. 1.70 GHz and 2 GHz Datasheet* or the *Intel ® Xeon ™ Processor MP at 1.40 GHz, 1.50 GHz and 1.60 GHz Datasheet,* and chipset datasheet)
- $T_{skew,\ max}$ = 0.59 ns.

Source Synchronous Signal Setup Time to Processor BCLK

The processor has a setup time requirement for the first transfer of source synchronous signals to the BCLK. See the *Intel® Xeon™ Processor at 1.40 GHz, 1.50 GHz. 1.70 GHz and 2 GHz Datasheet* and the *Intel ® Xeon ™ Processor MP at 1.40 GHz, 1.50 GHz and 1.60 GHz Datasheet* for details and diagrams on this timing requirement. This is only a requirement when an agent is driving to a processor. Use the same equation as the Common Clock signal setup time (Equation 9 below), but with the specific timings as specified by the *Intel® Xeon™ Processor at 1.40 GHz, 1.50 GHz. 1.70 GHz and 2 GHz Datasheet* and the *Intel ® Xeon ™ Processor MP at 1.40 GHz, 1.50 GHz and 1.60 GHz Datasheet*.

$$T_{margin\_setup} = T_{cycle,\min} - T_{co,\max} - T_{setup,\max} - T_{flight,\max} - T_{skew\_setup,\max} - T_{ssc,\max}$$

Equation 9 Source Sync. Signal Setup to BCLK Margin

Where:

- T_{cycle} is the minimum cycle time
- T_{skew_setup} is the skew between outputs from the clock driver and clock propagation skew on the PCB.
- T_{co} is the delay of the output buffer for the driving device
- T_{flight} is the interconnect delay specific to the platform.
- T_{setup} is the setup time required by the receiving buffer
- T_{ssc} is the jitter contributed by spread spectrum clocking

The timing parameters needed to complete this equation are as follows. Timings other than those provided in the component specifications are assumptions that are based on expected platform performance.

- T_{co} & T_{setup} are specified in the respective component specifications (chipset datasheet & *Intel® Xeon™ Processor at 1.40 GHz, 1.50 GHz. 1.70 GHz and 2 GHz Datasheet* or the *Intel ® Xeon ™ Processor MP at 1.40 GHz, 1.50 GHz and 1.60 GHz Datasheet*).
- T_{cycle} = 9.85 ns.
- T_{skew_setup} = 0.45 ns.
- T_{ssc} = 0.05 ns.

Common Clock

Setup Time

Equation 10 gives the equation for a common clock synchronous setup timing requirement.

$$T_{margin\_setup} = T_{cycle,\min} - T_{co,\max} - T_{setup,\max} - T_{flight,\max} - T_{skew\_setup,\max} - T_{ssc,\max}$$

Equation 10. Common Clock Setup Margin

The timing parameters needed to complete this equation are as follows. Timings other than those provided in the component specifications are assumptions that are based on expected platform performance.

- T_{co} & T_{setup} are specified in the respective component specifications (*Intel® Xeon™ Processor at 1.40 GHz, 1.50 GHz. 1.70 GHz and 2 GHz Datasheet* or the *Intel ® Xeon ™ Processor MP at 1.40 GHz, 1.50 GHz and 1.60 GHz Datasheet*, and chipset datasheet)
- T_{cycle} = 9.85 ns.
- T_{skew_setup} = 0.45 ns.
- T_{ssc} = 0.05 ns.

Hold Time

Equation 11 gives the equation for a common clock synchronous setup timing requirement.

$$T_{margin\_hold} = T_{co,\min} + T_{flight,\min} - T_{hold,\max} - T_{skew\_hold,\max}$$

Equation 11. Common Clock Hold Margin

Where:

- T_{skew_hold} is the skew between outputs from the clock driver and clock propagation skew on the PCB.
- T_{co} is the delay of the output buffer for the driving device
- T_{flight} is the interconnect delay specific to the platform.

- T_{hold} is the hold time required by the receiving buffer

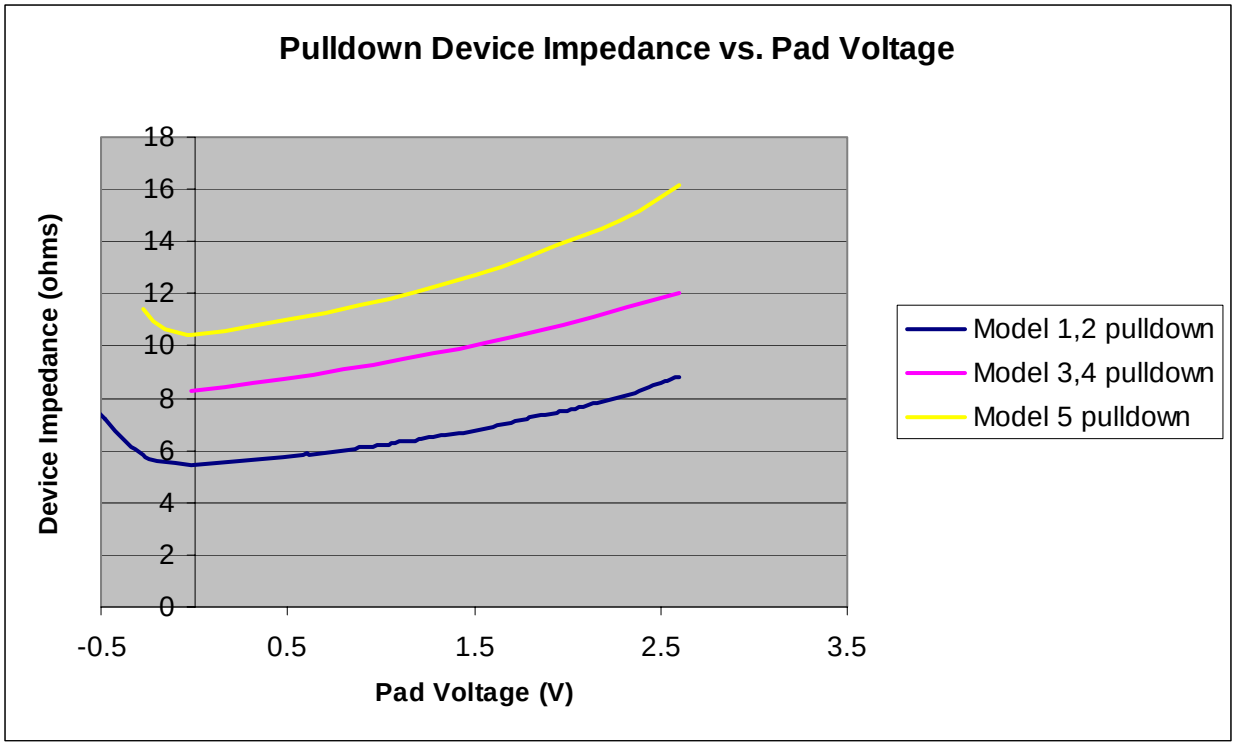
The timing parameters needed to complete this equation are as follows. Timings other than those provided in the component specifications are assumptions that are based on expected platform performance.

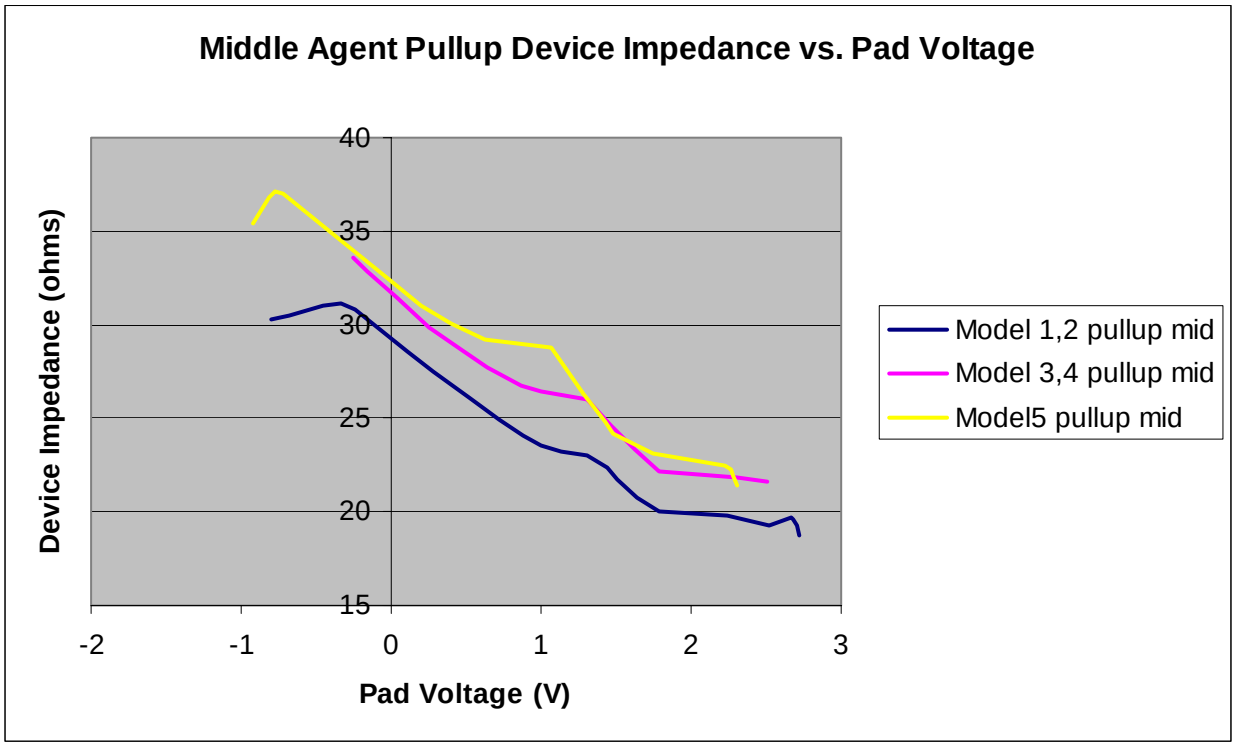
- T_{co} & T_{hold} are specified in the respective component specifications (*Intel® Xeon™ Processor at 1.40 GHz, 1.50 GHz. 1.70 GHz and 2 GHz Datasheet* or the *Intel ® Xeon ™ Processor MP at 1.40 GHz, 1.50 GHz and 1.60 GHz Datasheet*, and chipset datasheet)
- T_{skew_hold} = 0.45 ns.

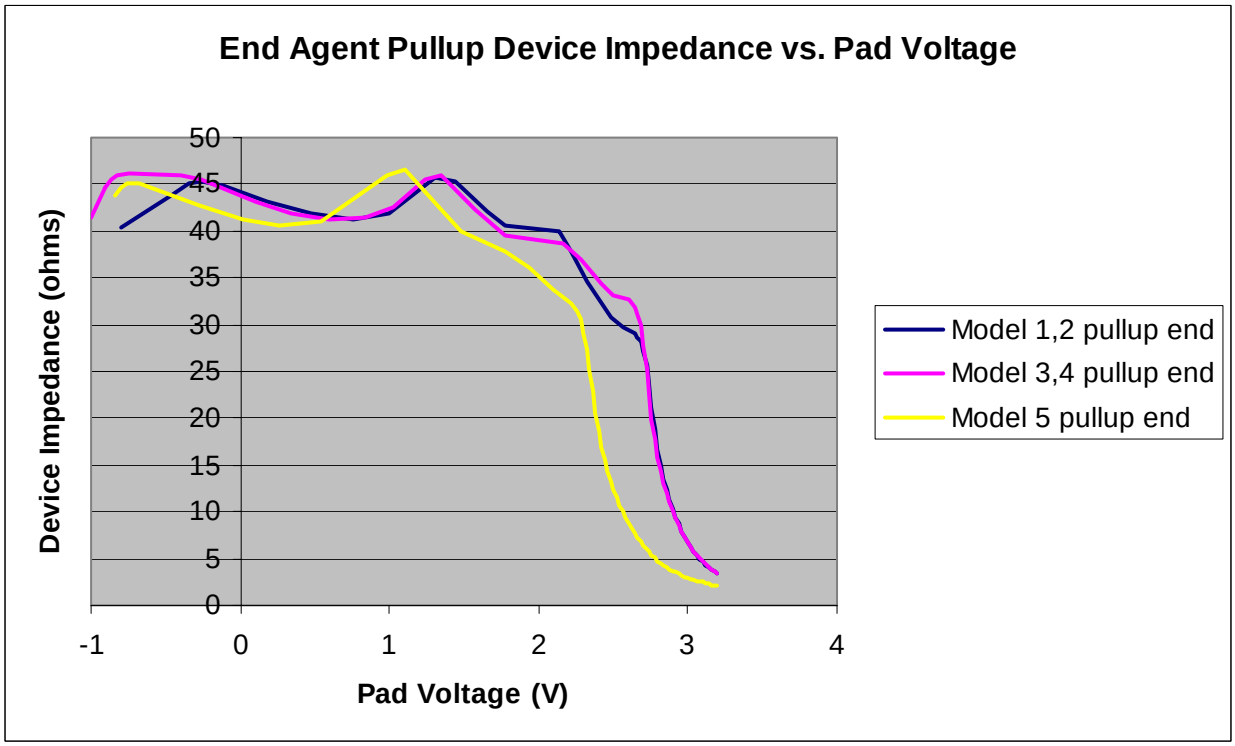
3.7 Model Characterization

Model Parameters							
Model (end and middle)	Pulldown	Pullup	rising edge rate	falling edge rate	Vcc	Temperature	Ceff
Model 1*	strong	strong	fast	fast	1.80	0° C	min
Model 2*	strong	strong	fast	fast	1.80	0° C	max
Model 3*	typical	weak	fast	fast	1.80	0° C	min
Model 4*	typical	weak	fast	fast	1.80	0° C	max
Model 5	weak	typical	slow	slow	1.50	100° C	max

* Models (1,2) and (3,4) contain the same device VI and VT data. Only Ceff is altered.
NOTE: The data is this table is a summary of the data provided in the follow plots.







4.0 Validation

These models were validated using Viewlogic* XTK Ver 6.4. Each model has been validated to ensure proper operation. Model data is consistent with Intel's most current data at time of creation. Note that these models are at revision 0.5 level, which means the data content is still based on pre-layout simulation data.

5.0 Package Model Information

5.1 Model Description:

The processor signal integrity package model consists of two transmission line segments representing traces routed in an OLGA package. One segment represents the trace routing before fanout and the other represents the trace routing after fanout. It also consists of the die bump parasitics, the PTH via parasitics, and the package land parasitics. All of these pieces are displayed in Figure 4 that represents the "Generic Package I/O Model". All of these models are symmetric stripline as per processor design rules. Also, each segment consists of 5 coupled transmission line traces and include both DC and frequency dependent losses (skin effect). The L's and C's were generated by solving 3D models constructed using Ansoft* Maxwell Q3D*. An example of this type of model is displayed in Figure 5. The L was solved at high frequency (i.e., all current was assumed to flow on the surface of the conductor and the capacitance was assumed to be frequency independent in the frequency range of concern). The losses were generated by solving 2D models, which were constructed using Viewlogic XFX*. The following models are provided to address performance metrics such as impedance and cross-talk: Nominal Model, Lowest Zo Model, Highest Zo Model, and Maximum Cross-talk Model. The models provided for each metric were skewed statistically to their corresponding 4 sigma values. The matrices for all of the models are distributed transmission line parameters and are provided in HSPICE* W element format in the .txt files.

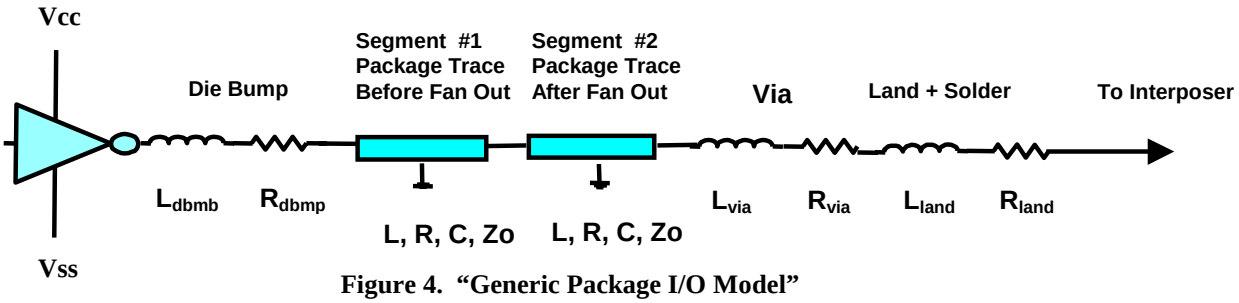


Figure 4. "Generic Package I/O Model"

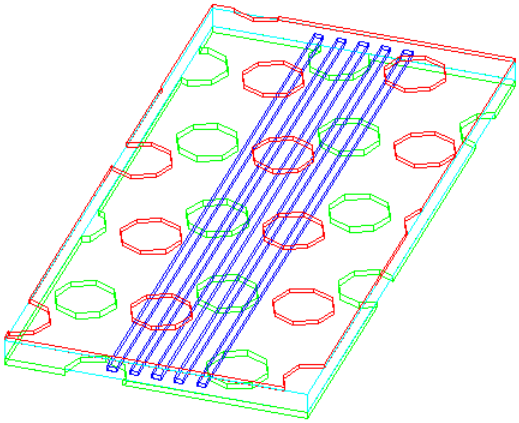


Figure 5. 3D View of 5 Coupled Traces

5.2 Model Limitations/Assumptions :

1. This model assumes ideal ground planes for the Vss and Vcc planes.
2. These models assume symmetric stripline throughout.
3. Manufacturing tolerances apply to transmission line segments only
4. The high frequency L was used and C was assumed to be frequency independent.
5. Assumes Offset 22.5 degree rotated degassing hole pattern

5.3 Model Usage:

It is highly recommended that the full 5x5 matrices be used.

1. The total length of the trace before fanout and after fanout is equal to the package trace lengths provided in the Length Adjustment Worksheet.
2. Trace length before fanout is 80 mils.
3. Trace length after fanout is equal to the total package length (from the spreadsheet) minus the length before the fanout (80mils).
4. Vary the type of model (Nominal, Low Zo, High Zo, and Maximum Cross-talk) as appropriate for the corner under simulation.

The parasitics for all model types are given in the next section.

The overall model is the "Generic Package I/O Model" which was displayed in Figure 4. The user can simulate the different conditions by using the values mentioned in Table 1 for the various pieces of this model. This yields 4 conditions that can be examined along with the different combinations of trace lengths for each.

Model Type	Segment #1	Segment #2	L_{dbmp} pH	R_{dbmp} m	L_{via} pH	R_{via} m	L_{land} pH	R_{land} m
Nominal	cpu_1	cpu_2	20	2	520	2.1	148	6
Low Zo	cpu _1	cpu_3	20	2	470	2.1	128	6
High Zo	cpu_1	cpu_4	20	2	570	2.1	168	6
Maximum Cross-talk	cpu_5	cpu_6	20	2	520	2.1	148	6

Table 1: Modeling values for all model types

5.4 Package Model File Details

The actual models are described below. Also below in Figure 6 the actual geometry cross section is displayed. The corresponding values for each of the parameters identified in this figure are listed in Table 2

The configuration in file "cpu_1.txt" contains parasitics for the trace scenarios before fan-out (trace segment near the die), giving nominal trace characteristic impedance where the geometrical parameters are nominal. (See Table 2 for geometry information)

The configuration in file "cpu_2.txt" contains parasitics for the trace scenarios after fan-out (trace segment far away from the die), giving nominal trace characteristic impedance where the geometrical parameters are nominal. (See Table 2 for geometry information)

The configuration in file "cpu_3.txt contains parasitics for the trace scenarios after fan-out giving low characteristic impedance. (See Table 2 for geometry information)

The configuration in file "cpu_4.txt" contains parasitics for the trace scenarios after fan-out giving high characteristic impedance. (See Table 2 for geometry information)

The configuration in file "cpu_5.txt" contains parasitics for the trace scenarios before fan-out giving the maximum cross talk. (See Table 2 for geometry information)

The configuration in file "cpu_6.txt" contains parasitics for the trace scenarios after fan-out giving the maximum cross talk. (See Table 2 for geometry information)

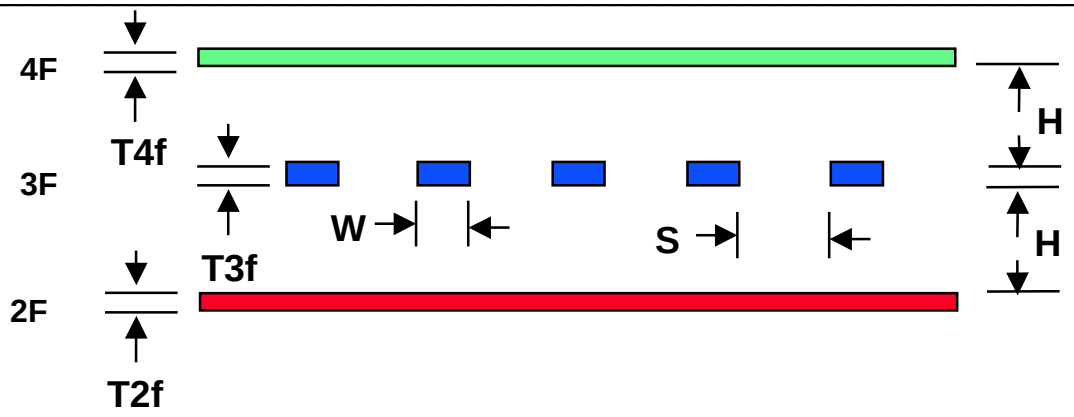


Figure 6. Cross-Sectional Geometry for All Model Types

Model	Plane Thickness Bottom T2f (µm)	Plane Thickness Top T4f (µm)	Trace Thickness T3f (µm)	Trace Width W (µm)	Dielectric Thickness H (µm)	Spacing S (µm)
Nominal (before fanout)	25	17	17	37	33	37
Nominal (after fanout)	25	17	17	37	33	54
Low Zo	25	17	17	42.6	25.5	48.4
High Zo	25	17	17	31.4	40.5	59.6
Maximum xtalk (before fanout)	25	17	17	41.22	40.5	32.78
Maximum xtalk (after fanout)	25	17	17	41.22	40.5	49.78

Table 2. Model Dimensions

5.5 Model Units:

The units for the data given in these files are:

HSPICE W element "cpu_x.txt"

L = H/m
C = F/m
Rdc = /m
Rs = /(m*Hz^.5)
Gs = S/(m*Hz)

6.0 Disclaimer and Legal Information

The *Intel® Xeon™ Processor (DP and MP) Signal Integrity Models* are solely intended for design validation and do not characterize tester guard-band or full process variation. They are not guaranteed to predict actual or future device behavior. This model suite is not a formal Intel specification and is intended to be a companion to official documentation. In all instances, Intel documented specifications override model predictions.

THESE MODELS ARE PROVIDED "AS IS" WITH NO WARRANTIES OR GUARANTEES WHATSOEVER. Intel EXPRESSLY DISCLAIMS ALL WARRANTIES AND GUARANTEES, INCLUDING ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS, AND ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

*Other names and brands are the property of their respective owners.